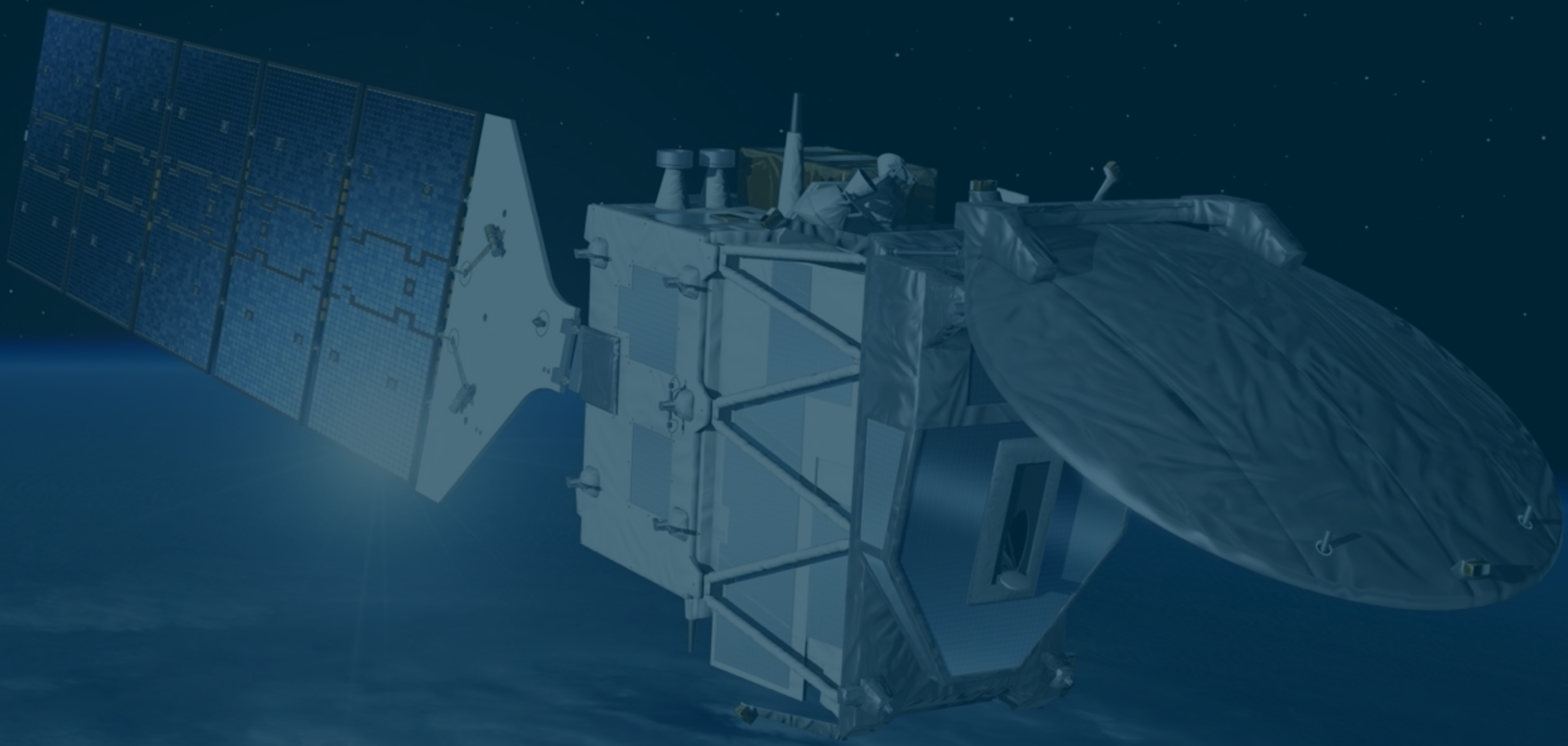


Atmosphere Virtual Lab (AVL)

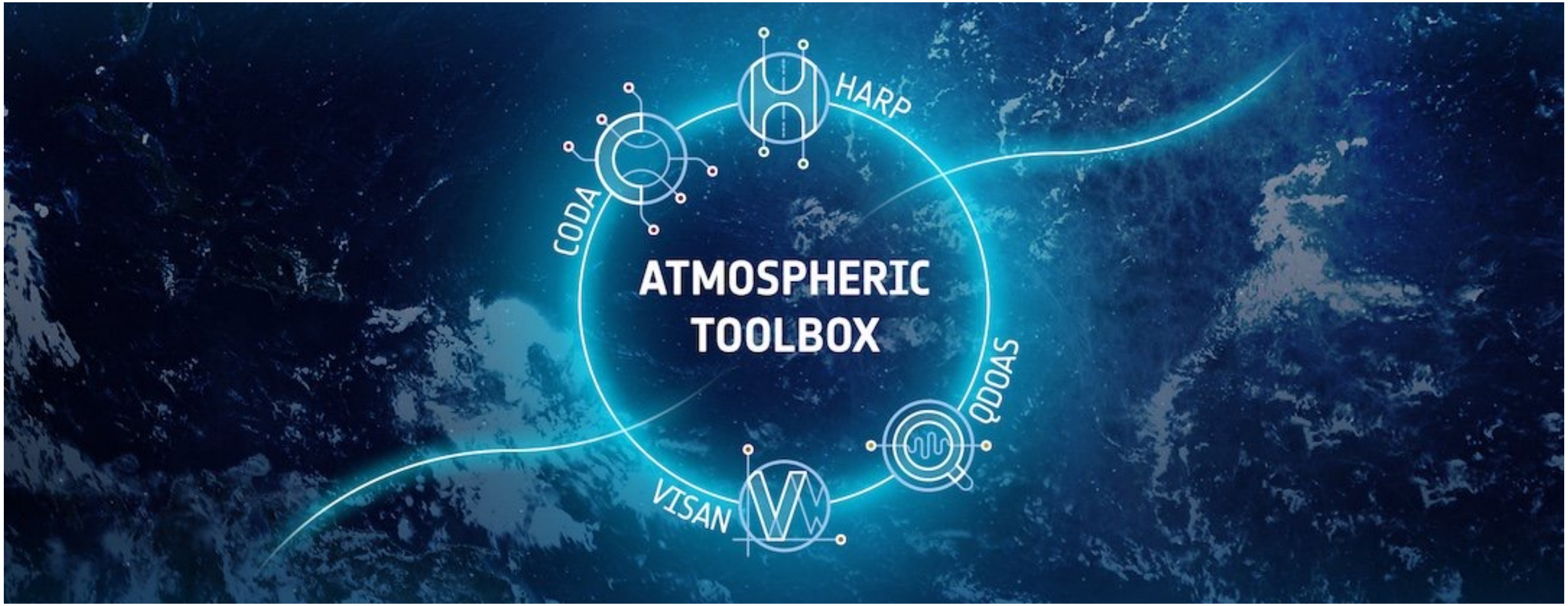
2nd ESA EarthCARE Validation Workshop

Sander Niemeijer, S[&]T

25-28 May 2021 (online)

Atmosphere Virtual Lab

- Continuation and evolution of the ESA Atmospheric Toolbox



Atmosphere Virtual Lab

- Atmospheric Toolbox consists of:
  - CODA – file reading library/tools supporting large variety of data formats
  - **HARP – library/tools for easy ingestion, data comparison, regridding, etc.**
  - QDOAS – DOAS retrieval application
  - VISAN – desktop application for visualisation

- See also presentation on ESA Atmospheric Toolbox (ATS) at the first EarthCARE validation workshop (13 June 2018)

https://atmospherictoolbox.org

Atmosphere Virtual Lab

- Focus in AVL project is on **bringing the tools to cloud platforms** (e.g. DIAS, Exploitation Platforms)
  - Most cloud platforms focus on Python and Notebooks for interactive analysis
- Also allow running AVL standalone on local computers
- Solution will be based on **JupyterLab**
  - Use what is already there (numpy, matplotlib, cartopy, plotly, etc.)
  - Improve/extend python interfaces for CODA/HARP/QDOAS
  - **Improve interactive geographical plots for atmospheric data** (replace VISAN as main visualisation environment)
- Project result will be software that can be deployed in exploitation platforms; AVL is not a cloud service itself

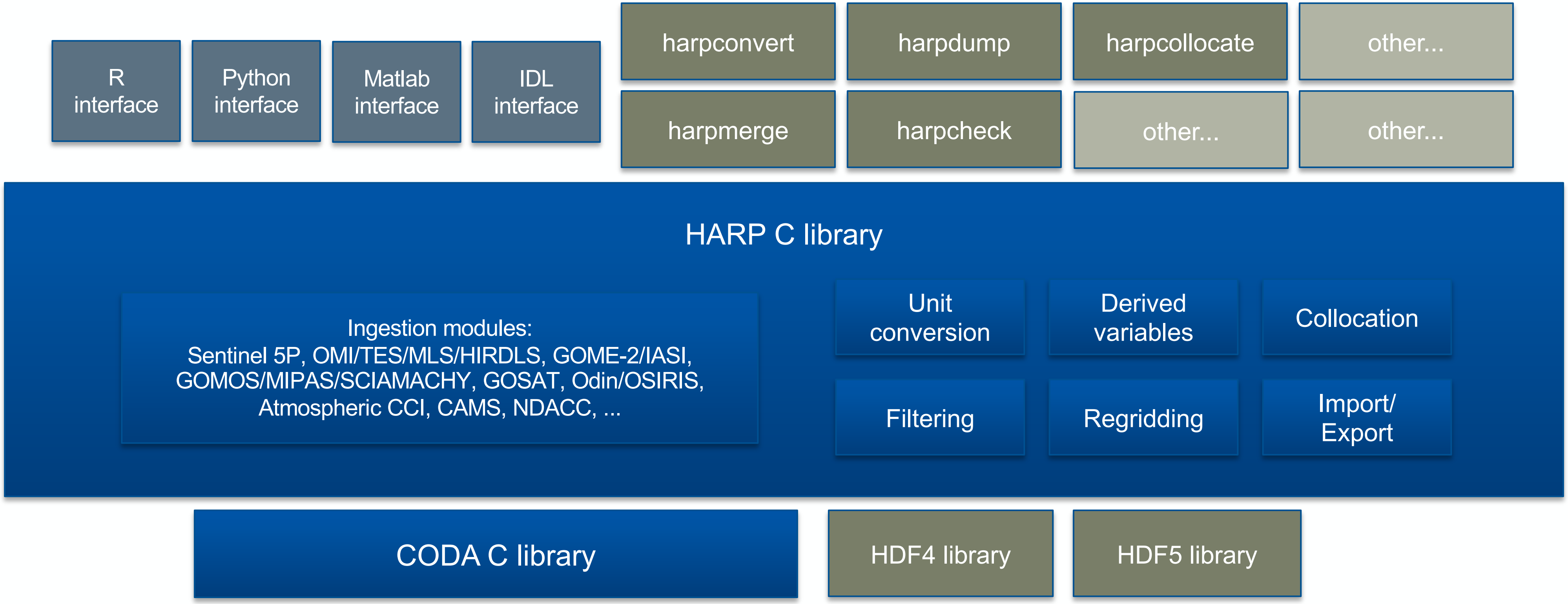
Atmosphere Virtual Lab

- Besides the Lab environment, the AVL project also covers:
  - support for new missions such as EarthCARE
  - Create publicly available Use Cases and Examples

- Project only started recently
- No support for EarthCARE yet, but coming soon

Atmosphere Virtual Lab

- Most important component for validation community is **HARP**



Atmosphere Virtual Lab

- HARP is not just software, but also a set of conventions:
  - simple data structure like netcdf (product with variables; no grouping)
  - global attributes, variable attributes
  - similar/complementary to e.g. netCDF-CF, but not fully compatible (CF couples dimensions to coordinate variables; HARP keeps these fully separate)
  - variable naming convention
    - similar things should be named similarly
    - different things should be named differently
    - measurements from different instruments or different retrieval algorithms belong in different files (allows use of same variable name)

Atmosphere Virtual Lab

```
[1]: import harp
     product = harp.import_product("S5P_OFFL_L2__CLOUD__20210521T144947_20210521T163117_18674_01_020104_20210523T074128.nc")
     print(product)
```

```
source product = 'S5P_OFFL_L2__CLOUD__20210521T144947_20210521T163117_18674_01_020104_20210523T074128.nc'

int scan_subindex {time=1877400}
double datetime_start {time=1877400} [seconds since 2010-01-01]
float datetime_length [s]
int orbit_index
long validity {time=1877400}
float latitude {time=1877400} [degree_north]
float longitude {time=1877400} [degree_east]
float latitude_bounds {time=1877400, 4} [degree_north]
float longitude_bounds {time=1877400, 4} [degree_east]
float sensor_latitude {time=1877400} [degree_north]
float sensor_longitude {time=1877400} [degree_east]
float sensor_altitude {time=1877400} [m]
float solar_zenith_angle {time=1877400} [degree]
float solar_azimuth_angle {time=1877400} [degree]
float sensor_zenith_angle {time=1877400} [degree]
float sensor_azimuth_angle {time=1877400} [degree]
float cloud_fraction {time=1877400} []
float cloud_fraction_uncertainty {time=1877400} []
byte cloud_fraction_validity {time=1877400}
float cloud_fraction_apriori {time=1877400} []
float cloud_base_pressure {time=1877400} [Pa]
float cloud_base_pressure_uncertainty {time=1877400} [Pa]
float cloud_base_height {time=1877400} [m]
float cloud_base_height_uncertainty {time=1877400} [m]
float cloud_top_pressure {time=1877400} [Pa]
float cloud_top_pressure_uncertainty {time=1877400} [Pa]
float cloud_top_height {time=1877400} [m]
float cloud_top_height_uncertainty {time=1877400} [m]
float cloud_top_temperature {time=1877400} [K]
float cloud_optical_depth {time=1877400} [m]
float cloud_optical_depth_uncertainty {time=1877400} [m]
byte cloud_type {time=1877400}
float surface_albedo {time=1877400} []
float surface_albedo_uncertainty {time=1877400} []
float surface_altitude {time=1877400} [m]
float surface_altitude_uncertainty {time=1877400} [m]
float surface_pressure {time=1877400} [Pa]
float surface_meridional_wind_velocity {time=1877400} [m/s]
float surface_zonal_wind_velocity {time=1877400} [m/s]
byte snow_ice_type {time=1877400}
float sea_ice_fraction {time=1877400} []
long index {time=1877400}
```

Atmosphere Virtual Lab

- HARP variable naming convention
  - Already proper distinction for e.g.:
    - densities (total column, partial column profile, concentration profile)
    - number density vs. mass density (also for mixing ratios)
    - total air mixing ratios vs. dry air mixing ratios
  - Variables names are generic, and sometimes differ from domain specific terms.
    - e.g. a vertical ‘water path’ would become a ‘column density’.
  - **Cloud/Aerosol domain still work in progress**. Complicated due to water phases (vapor, liquid, ice), precipitation types, cloud phases, aerosol types/sizes, etc.

http://stcorp.github.io/harp/doc/html/conventions/index.html

Atmosphere Virtual Lab

- HARP operations
  - filtering
  - derivations
  - vertical regridding
  - binning
  - spatial binning
  - applying averaging kernels
- Operations are provided as a ';' separated string parameter during ingestion of a file:

```
product = harp.import_product("S5P_OFFL_L2__CLOUD__20210521T144947_20210521T163117_18674_01_020104_20210523T074128.nc",
                              "cloud_fraction_validity>75;latitude>=0;latitude<=50;keep(latitude,longitude,cloud_top_height);"
                              "derive(cloud_top_height [km])")
```

Collocation workflow with HARP:

```
$ harpcollocate -d 'datetime 12 [h]' -d 'point_distance 500 [km]' -ny point_distance
    -nx datetime .../sat_data_dir .../ref_data_dir collocations.csv

$ harpmerge -a 'collocate_left("collocations.csv")' .../sat_data_dir sat_data.nc
$ harpmerge -a 'collocate_right("collocations.csv")' .../ref_data_dir ref_data.nc
```

You can then further manipulate the data (e.g. smoothing, profile -> column, ...)

```
$ harpconvert –a '.....' data.nc data_modified.nc
```

or:

```
>>> harp.import_product('data.nc', '…..')
```

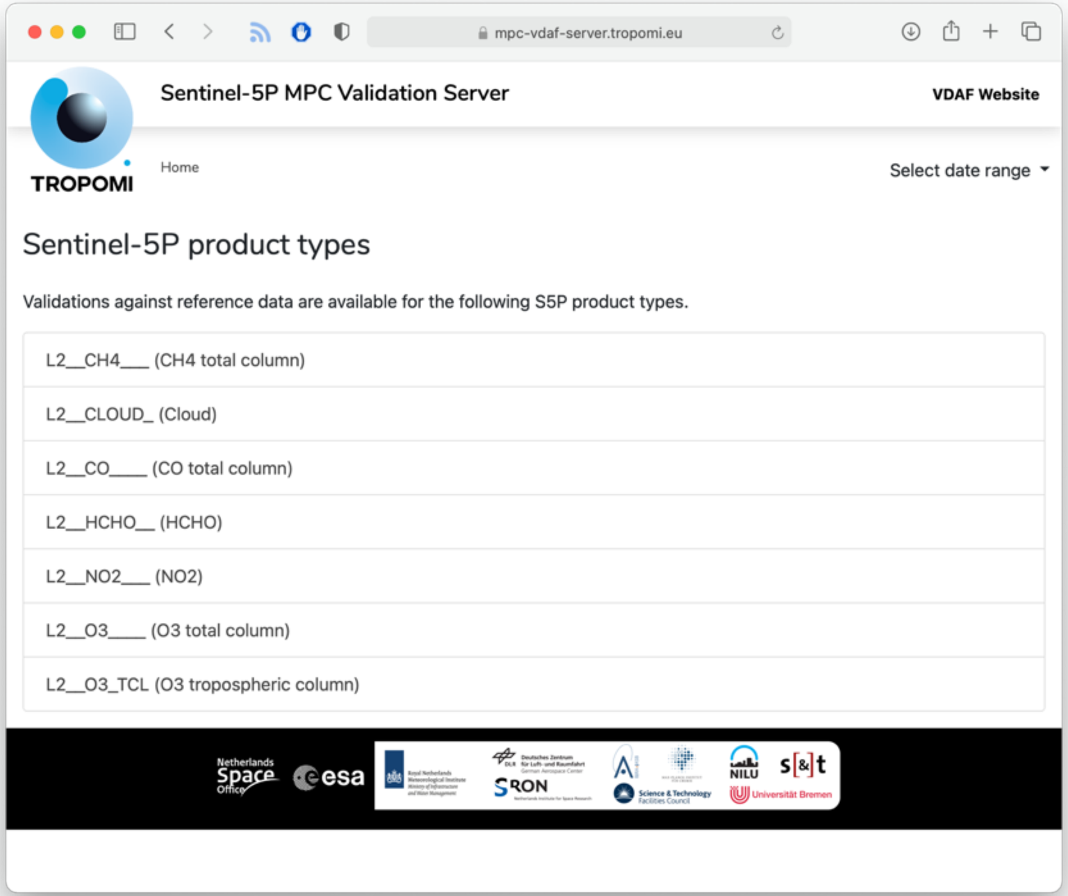
Atmosphere Virtual Lab

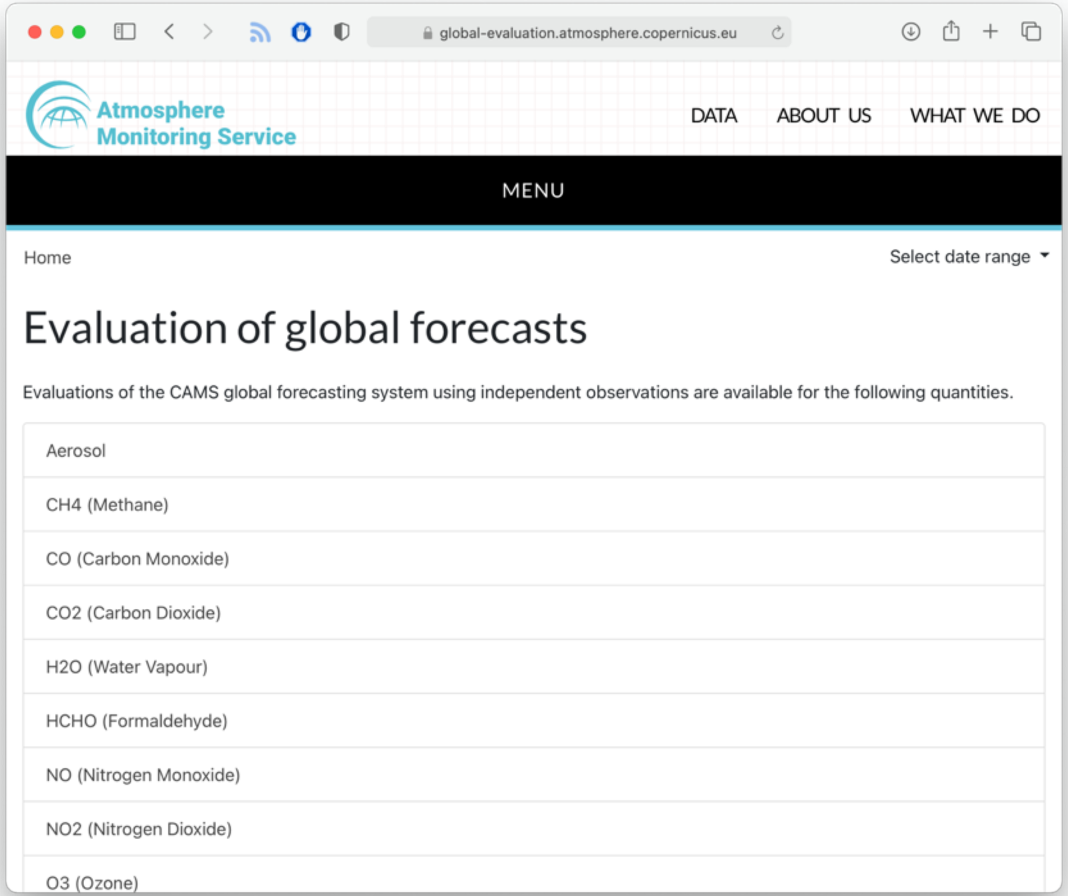
- Upcoming support for EarthCARE:
  - Ingestion support for all L2A and L2B products (main variables only) http://stcorp.github.io/harp/doc/html/ingestions/index.html
  - Ingestion support for reference data (e.g. EVDC, GEOMS, cloudnet, …)
  - Review/adapt HARP conventions for cloud/aerosol
  - Improve/revise HARP operations for cloud/aerosol domain

  - Support EarthCARE with AVL visualisation functions
  - Publish Use Cases / Examples for handling EarthCARE data with AVL/HARP

Atmosphere Virtual Lab

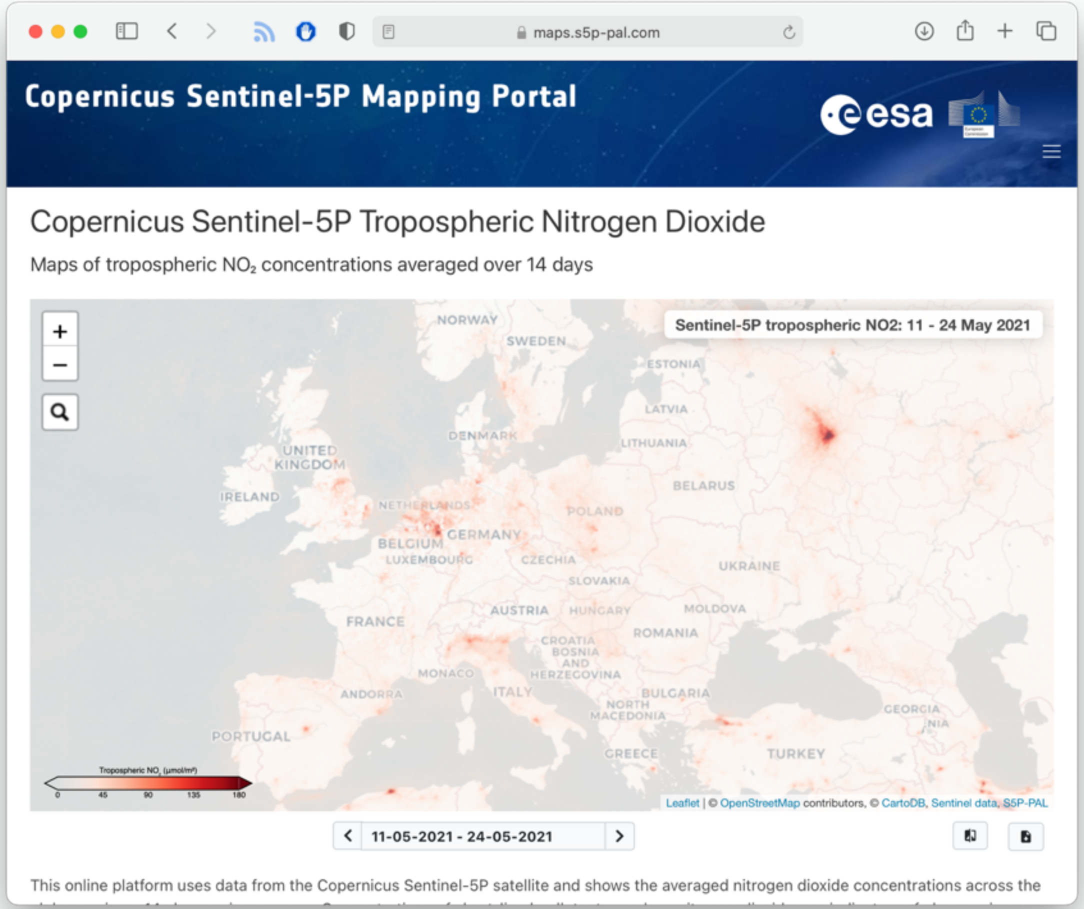
HARP is used in many operational services already:





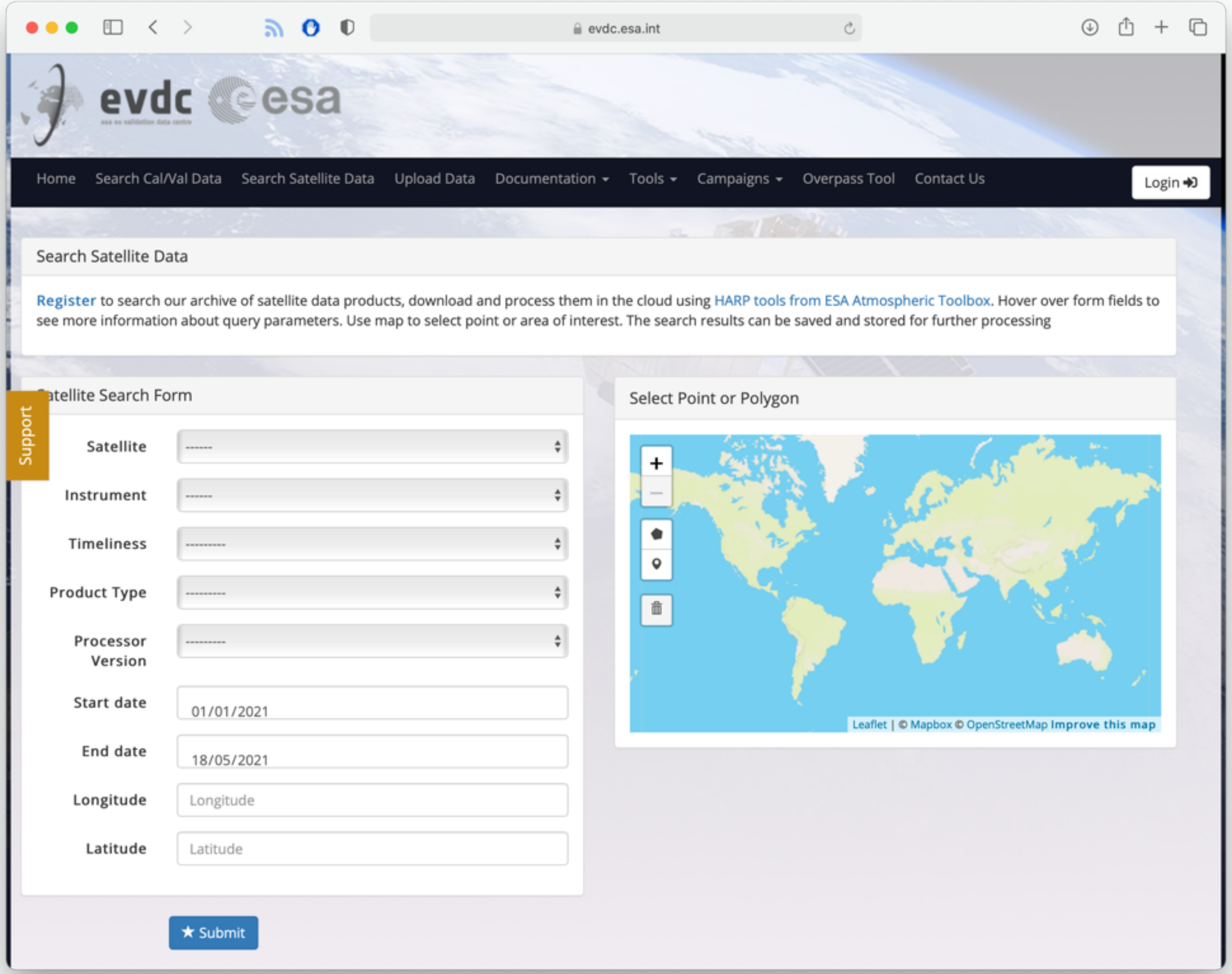
Atmosphere Virtual Lab

HARP is used in many operational services already:

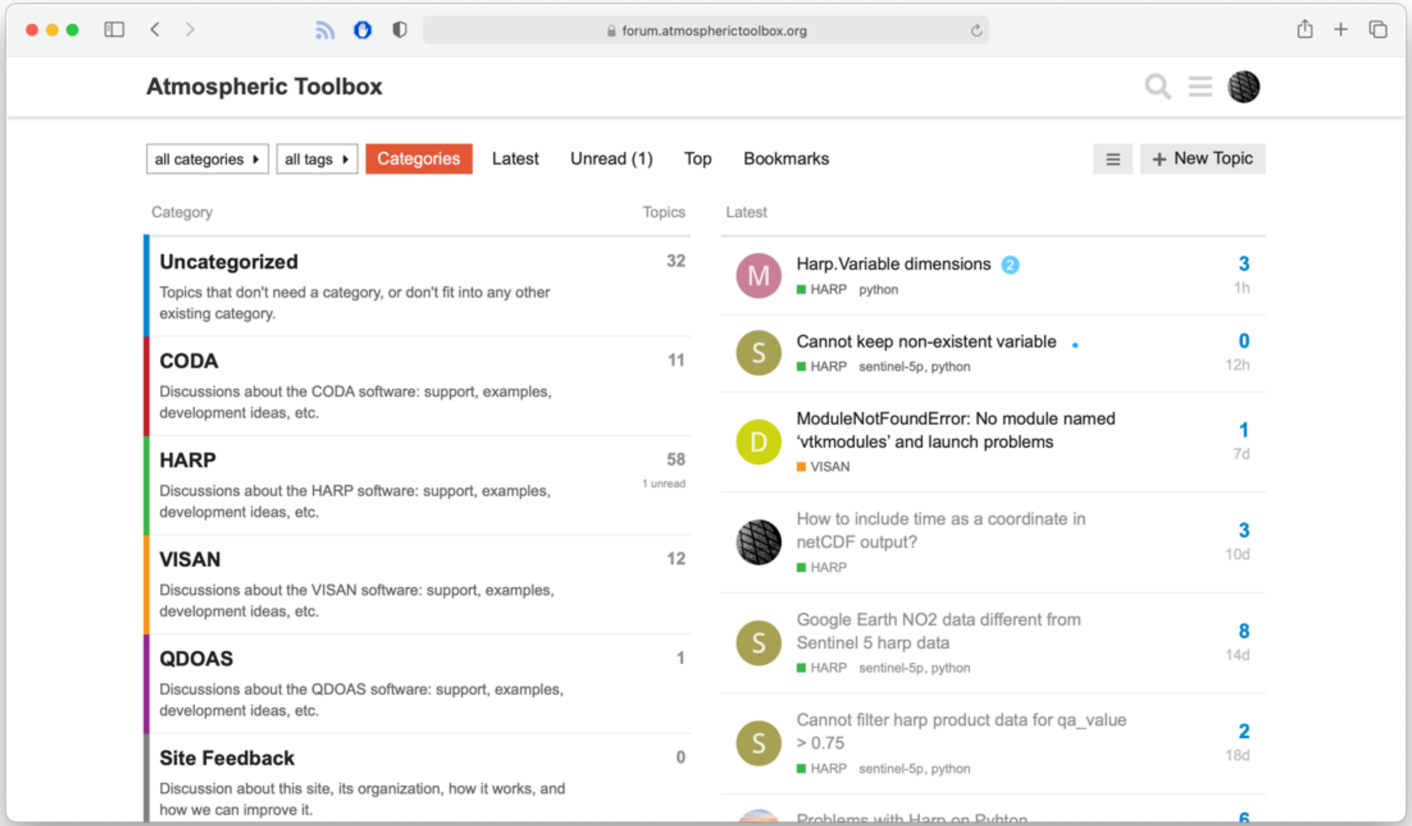


Atmosphere Virtual Lab

HARP is used in many operational services already:



Atmosphere Virtual Lab



https://forum.atmospherictoolbox.org